# Comp 3350: Computer Organization & Assembly Language

# HW # 8. Theme: Integer Arithmetic

*All main questions carry equal weight.*

*Points will be awarded to only those answers which have work clearly shown*

1. In the following code sequence, show the value of AL after each shift or rotate instruction has executed. This question is to be done by hand, not programmatically.

mov cl, 3

mov al, 45h

rol al, cl ; al = 2A

mov al, 7Ah

mov cl, 1

ror al, Cl ; al = 3D

stc

mov al, 64h

mov cl, 2

rcl al, cl ; al = 92

stc

mov al, 3Dh

mov cl, 1

rcr al, cl ; al = 9E

A close up of text on a white background

Description generated with very high confidence

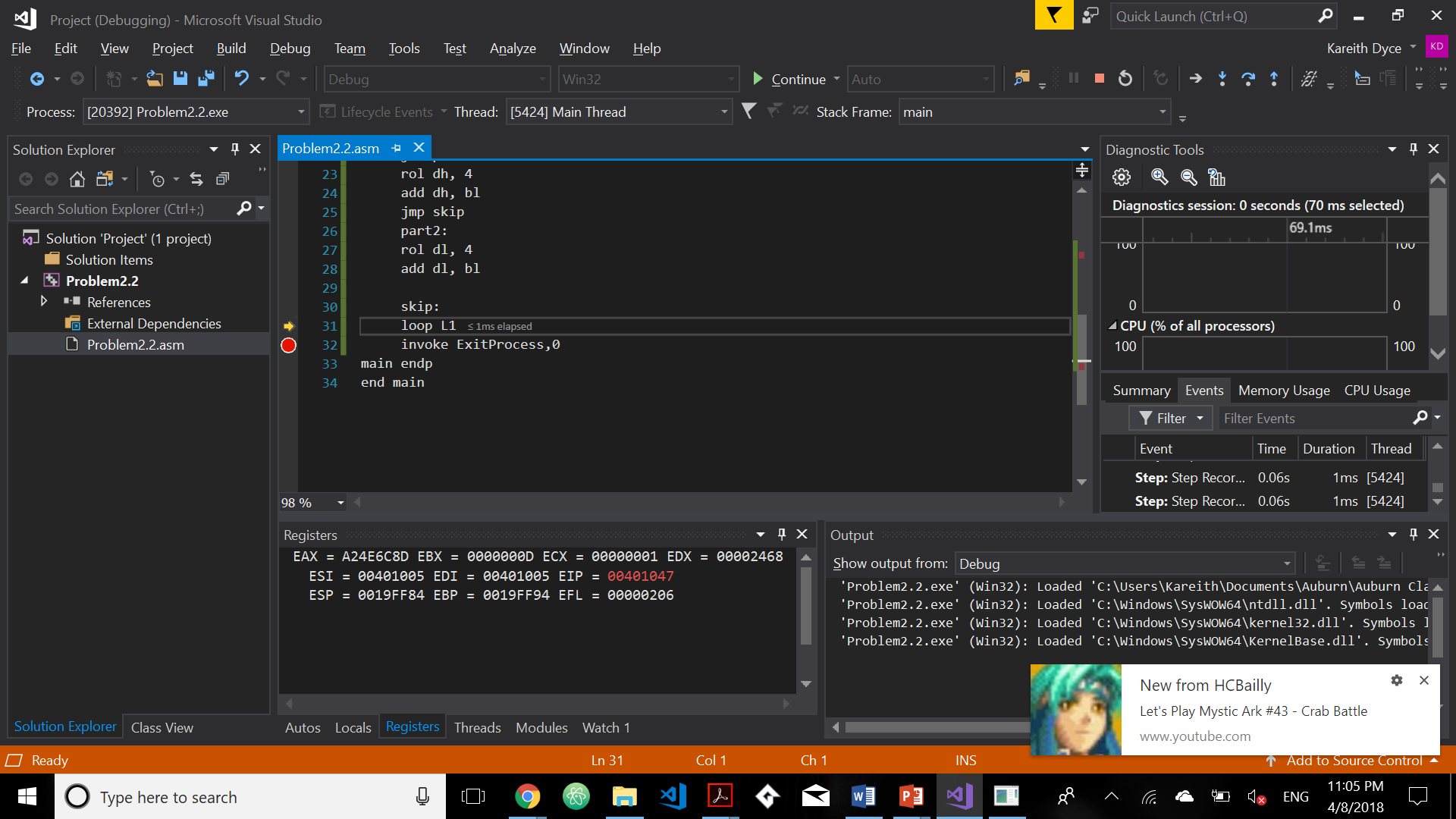
1. (a) Write a program that calculates EAX\**1710* using binary multiplication.

mov ebx, eax

shl ebx, 4

add eax, ebx

(b) Consider the following value: A24E6C8D. Let this value be stored in register EAX. Write a program that will extract the decimal digits from this value using shifts and logical instructions. Place the first two **decimal numeric** digits in DH and the other two into DL. Submit a print out of the run of the program and the list file.



1. (a) What will be the contents of AX and DX after the following operation? You must work this problem by hand, not by a program run. What may happen if you do not set dx to 0 in the beginning?

mov dx, 0

mov ax, 1234h

mov cx, 4213h

mul cx

AX = C1DC

DX = 04B2

If dx is not cleared it will be set after the mul instructionA close up of text on a white surface

Description generated with high confidence.

(b) When does an IDIV instruction cause an overflow? Provide an example. The IDIV instruction would cause an overflow if the reminder portion was not properly set up(cleared and then had its sign extended).

mov ax, -53h

mov bx, -5h

mov dx, 1h

IDIV bx

1. What will be the values of DX:AX after the following instructions execute? What might be the use of such a sequence of instructions in a 16-bit computer?

mov ax, 0h

mov dx, 0h

sub ax, 1h

sbb dx, 0

DX:AX= FFFF:FFFF

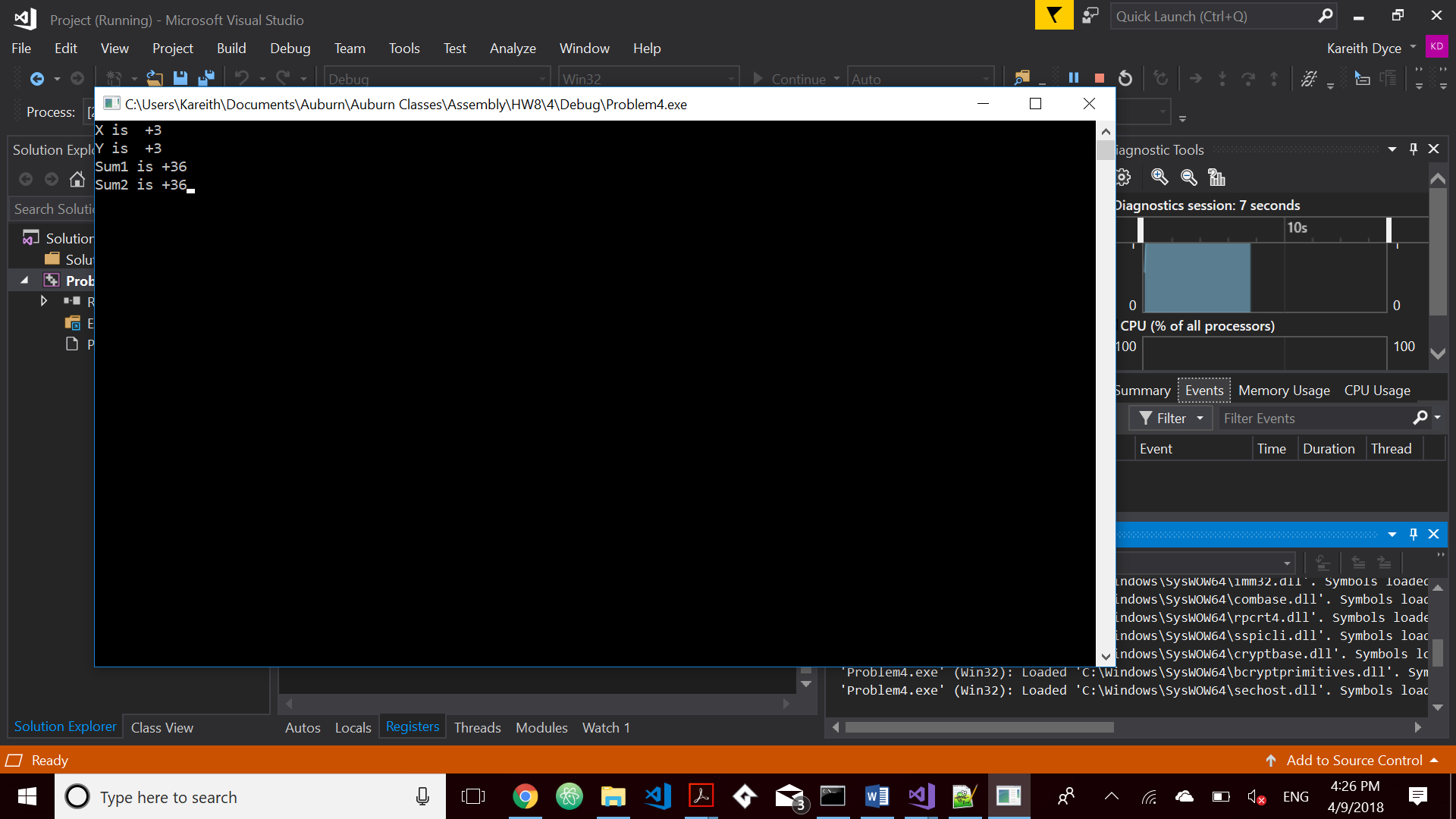
With these instructions we are able to carry out 32bit math using only 16 bit resources. This means our computers would be able to process and compute much more by combining 16 bit resources.

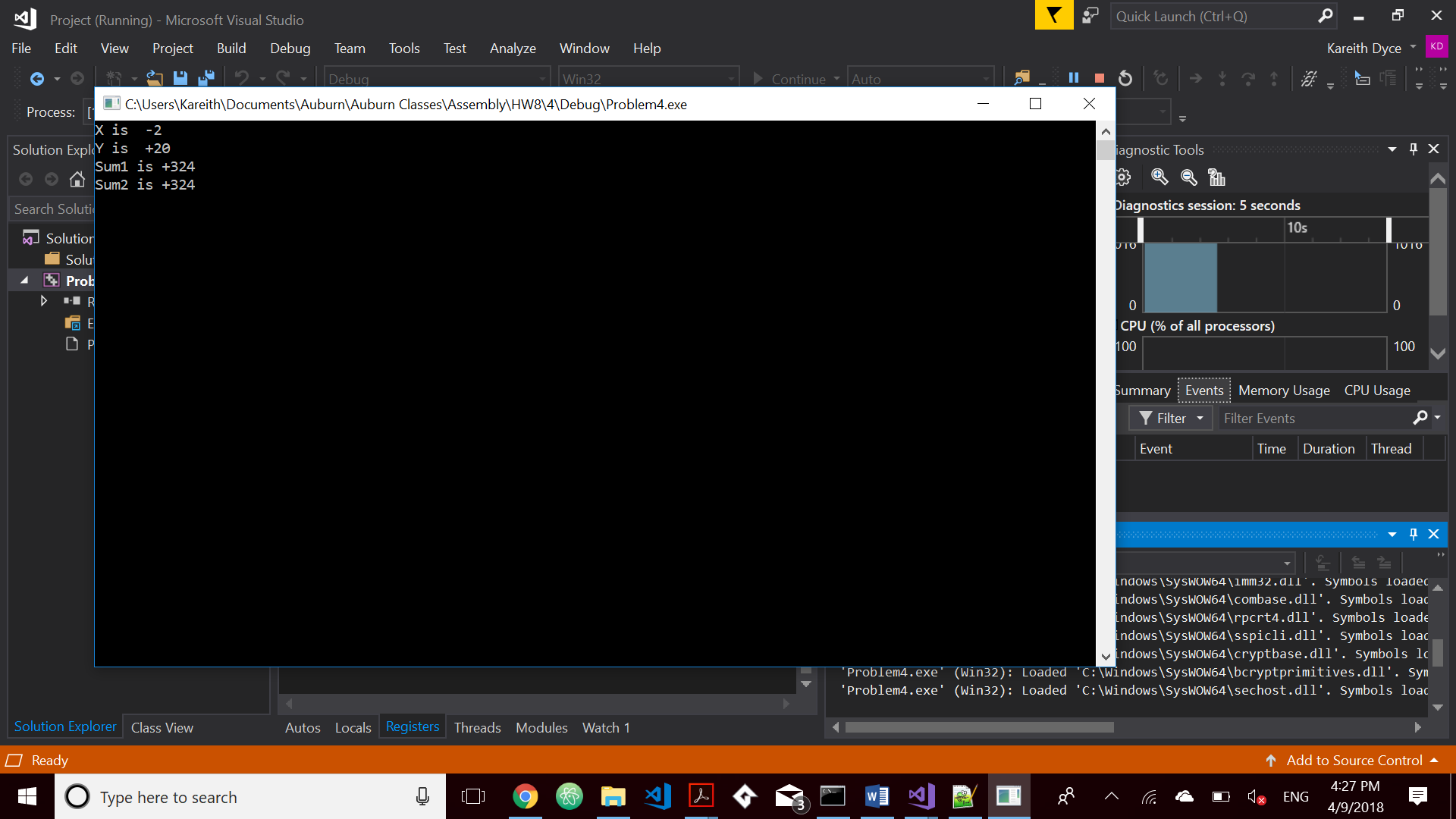
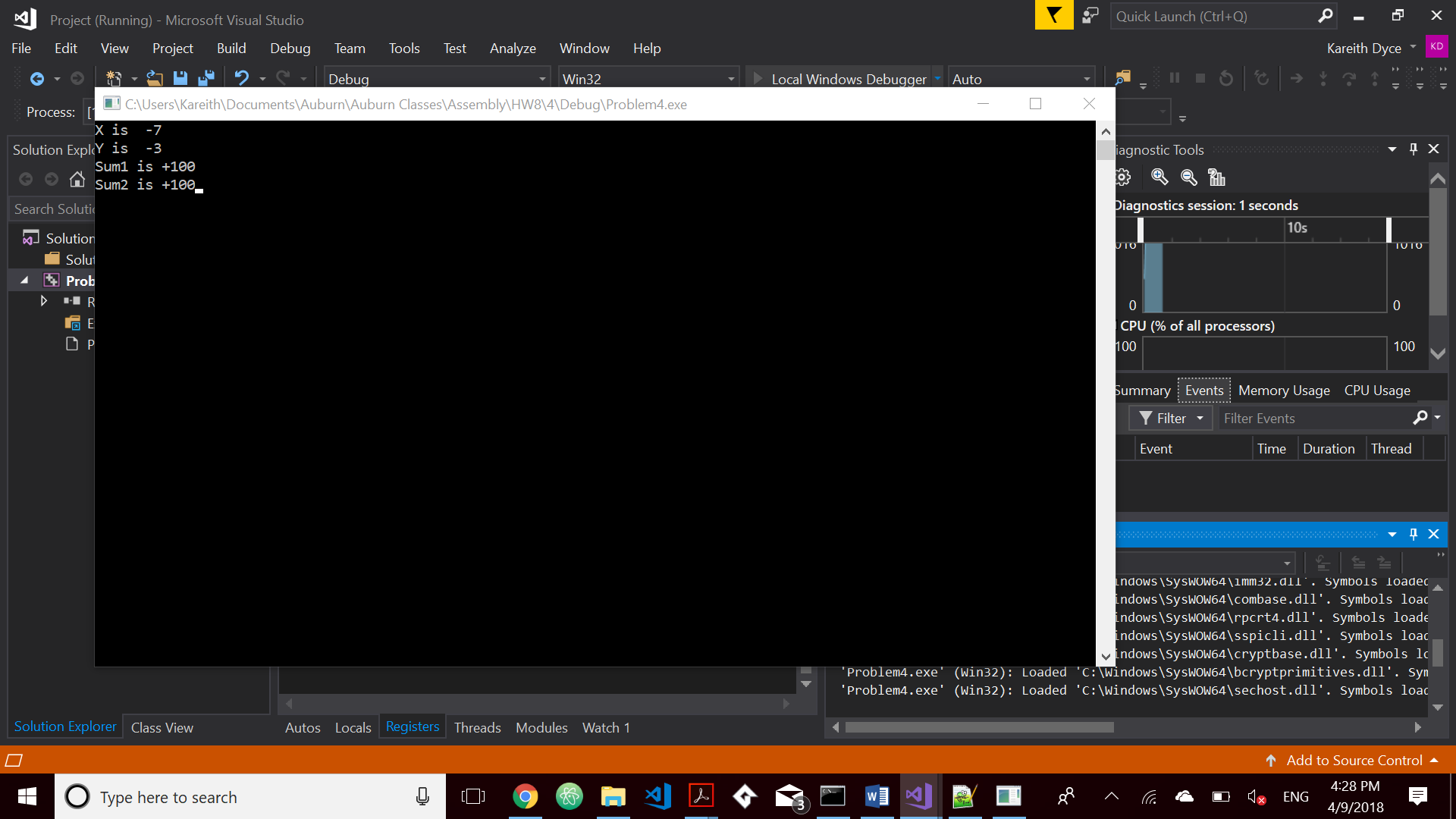
1. Implement the following two expressions in assembly language, using 32-bit signed operands. Demonstrate the equivalence of the two using some test values for X and Y. Show the runs of the programs using both positive and negative test values. Which of the two implementations is preferable?

*Z = X2 + 2XY + Y2*

*Z = (X+Y)2*

The second one as it requires less lines of code.



1. Write a program that performs C = A – B using extended subtraction. See textbook pg. 270-271.

Use the following:

Apple WORD 1214h, 3423h, 6578h, 5699h, 2005h

Berry WORD 4125h, 2345h, 12BCh, 0CDF1h, 1009h

Cherry WORD 5dup(0)

Submit the list file and a display of the contents of all the arrays after the run. 